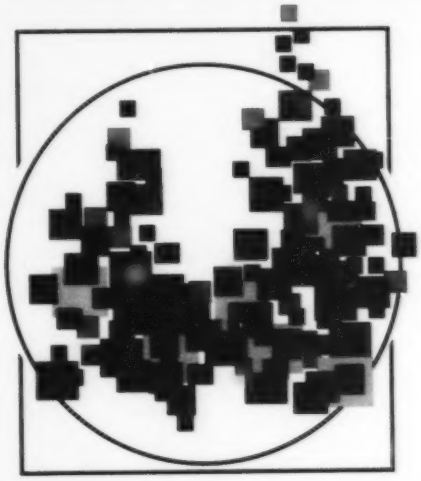

CENTRE FOR
SYSTEMS
SCIENCE

Simon Fraser University, Burnaby, B.C., Canada V5A 1S6 / (604) 291-4704

92-05741/

Natural Language Generation Systems and Theories

T Pattabhiraman and Nick Cercone

CSS-IS TR 90-10

December 1990

Natural Language Generation Systems and Theories

T. Pattabhiraman
Nick Cercone

CSS-IS TR 90-10

Centre for Systems Science
Simon Fraser University
Burnaby, B.C., Canada V5A 1S6
patta@cs.sfu.ca and *nick@cs.sfu.ca*
Tel#: (604)-291-3229

Abstract

Natural Language Generation (NLG) systems are constructed with a variety of research objectives such as systematic exploration of new research issues, computational modelling of scientifically-oriented theories of NLG and instantiation of descriptive theories from related disciplines like linguistics and philosophy of language. NLG theories and systems take widely different standpoints on a number of crucial theoretical and computational issues like the representation of linguistic knowledge, text planning and generator efficiency. Many of the issues are still unresolved, and connected to particular approaches and research objectives. Quantitative blackbox evaluation of NLG systems requires consensus on what concepts and results are critical to the problem, so that the goals of evaluation can be defined precisely. As a step towards devising metrics and procedures for evaluation, we distinguish NLG systems along a number of dimensions such as objectives, stages of evolution and approaches to the modelling linguistic knowledge. A classification of NLG systems thus obtained would aid in defining what sort of evaluation is appropriate to each kind, and would help relate some of the difficulties of evaluation to unresolved theoretical issues.

Keywords: Natural Language Generation, Methodology, Theory, System, Program, Evaluation

1. Introduction

In this paper we discuss theories of Natural Language Generation (NLG) from the viewpoint of how programming has aided in forming or instantiating theories, and how theories from related disciplines have been adapted in NLG.[1] We simultaneously address methodological, theoretical and evaluation issues in NLG. The goal of the paper is not to endorse or criticize any particular line of research or methodology, but rather to attempt an explanation of how the *diversity* of research objectives, methodologies and influences in NLG can be systematically reconciled and understood in a way that gives *unity* to the discipline, and how the understanding so obtained can be taken into account in devising procedures for evaluating NLG systems or theories.

What is Natural Language Generation?

- *Scope of the NLG problem*: The computational processes that take place from the speaker's decision to speak to the completion of the utterance form the subject matter of NLG [27]. As such, the temporal course of the NLG problem spans not only *tactical* generation of surface utterances, but also the *strategic* aspects of preparing representations for subsequent tactical generation. Not everyone explores the problem in its full scope; however, the scope of the full NLG problem is commonly accepted as such, and individual efforts, situated in the larger scheme of the full scope. Many issues of interest are not localized temporally to one stage of NLG: for example, pragmatic setting considered as a knowledge source influences strategic- as well as tactical aspects of NLG [12].

- *NLG as a discipline of enquiry*: As a field, NLG has included in its fold researchers from AI, computational linguistics, cognitive science, linguistics, psycholinguistics and other related branches, with common concerns for computational study of NLG. Accordingly, we discuss matters of import for computational models, and omit detailed discussions of empirical studies, psychological experimental methods and other techniques of the neighbouring disciplines.

- *NLG theories and models*: Theories of NLG are concerned with linguistic *performance* (as opposed to *competence*), and models of NLG are regarded as *performance models*. *NLG systems* are implemented as computational models of NLG or subprocesses thereof. An account of the theory typically exists detached from the program code of the system which is regarded as a *model* instantiating the theory. (For a discussion of the senses of *theory* and *model*, the reader is referred to Wilks [44] and references cited therein.)

- *NLG system*: An NLG system exudes different facets of its identity: as a generator of NL text, it has a metaphorical role as the speaker. As a system, it is an assembly of interconnected functional components. As a program, it is an implementation of algorithms and data structures. As an ancillary to or embodiment of a theory, it is a simulation model of a process, with both

---

[1]Portions of this report were presented as a position paper at the AAAI'90 Workshop on Evaluating Natural Language Generation Systems (Boston, 1990).

static properties of design and dynamic properties of behaviour at run-time being significantly connected to the theory. As in most branches and approaches of AI, a certain measure of ambivalence is generally maintained about whether the subject matter of enquiry is NLG by humans or by computers. Many research efforts aim at both. All of them allude to human NLG, and benefit from introspection, and from analytical, empirical or experimental studies of human NLG, if not *make claims* regarding human NLG.

Evaluation: Evaluation guides development in research, measures the contributions of a theory, determines its place among *rival* theories, helps assess competing products, and in general, canalizes research along potentially useful directions. The word *evaluation* means different things to different people, and in different contexts. The relevant dimensions along which evaluation is carried out, as well as the units in terms of which the results of evaluation are reported depend upon what is being evaluated and who is evaluating.

In the case of natural language generation (NLG), the following two contrasting categories of systems can be observed [14]:

1. As products, NLG systems serve as *mouthpieces* for intelligent host programs. In this case, they inherit many criteria and metrics from the evaluation methodology of software systems in general (for example, system-internal factors like response time, robustness, portability, error-handling and correctness, and system-external ones that involve *people-factors*- based on end-users' perception of quality).

2. As ancillaries to theorizing on NLG, they inherit many criteria and measures of evaluation from the general methodologies of theory assessment. Although many software-characteristics of such programs are relevant in the evaluation process, the status of these systems as pieces of software recedes to the background, giving prominence to their status as *models* of *theories*.

Although many dimensions of evaluations may be commonly applicable to both cases, the most relevant ones are distinct.

Midway between the above two categories, as it were, we can identify a third type of NLG systems, viz., applied (prescriptive) theories: We can distinguish between *descriptive* theories ((2) above) and *prescriptive* theories. Though both are executed as engineering projects, scientific concerns are prominent in the former, while engineering considerations can override psychological fidelity in the latter. As instances or embodiments of prescriptive theories, we may cite generators in machine translation systems, explanation systems and summarization systems.

In the rest of this paper we will present some issues pertinent to the theory-making value of NLG systems. Discussing these issues prior to designing evaluation dimensions, units and methodologies, to us, seems helpful in realizing the complexities of evaluation,

as well as in understanding how natural language generation has emerged as a research field. Section 2 is devoted to the role of programming in NLG, and some relevant methodological discussions from the perspectives of AI and cognitive science. Some aspects of the representation of linguistic knowledge in NLG systems are discussed in section 3. In section 4 we discuss evaluation issues. Our concluding remarks are presented in section 5.

2. Methodological and Metatheoretical Issues

The relationship between programs and theories has been explicitly discussed in AI and in cognitive science by influential researchers who have been concerned with questions of methodology and progress in the disciplines. Much of the criticism has been directed at work in which empirical credibility for theoretical claims was sought with the presence of a large program (let's call it the *AI program*) that performed impressively. Ritchie and Hanna [39], in their critique of *AM*, the mathematical discovery program, state that all aspects of a program that contribute to the theoretical claims of the authors should be well-documented. That is, the program should not have any hidden sources of power that fuelled its impressive performance. The problem is not that programming is theoretically sterile or that AI programs cannot contribute substantially to our knowledge. Rather, an undue amount of spotlight on the performance of the program, without a proportionate attention paid to explaining its internal workings, fails to provide the solid foundation for continuing research and accumulating knowledge. The mechanisms of the program interact in order to contribute to its performance. Theoretical principles underlie, or form the basis for, the mechanisms of the program. Therefore assessing the worth of a program implies nothing less than assessing its principles. In the light of the current interest in assessing NLG systems, the objection of Ritchie and Hanna to evaluating performance to the exclusion of examining system components and theoretical principles is worth noting.

Of course, it is hard for a researcher to judge which details to report and which ones to omit. Giving a full program listing would make a report painfully dull [21]. Nor is it feasible or reasonable to check in detail the text of a complex program and its runtime behaviour, and infer the theory from the program code and its execution [39], [31]. Errors of omission in reports are common and could be inadvertent. Moreover, programs are nothing but implementations of data structures and algorithms, on which a behavioural interpretation is superimposed by the theorizer/programmer. Since it cannot be proved rigorously that a program embodies a given principle, the best that one can do is to justify with carefully-articulated arguments [35]. In order to do so, it is crucial to separate the *theoretical principles* from the *technical details* that support instantiating those principles in programs, and from the idiosyncrasies of the programming language. As important as it seems, this is by no means a trivial task for large programs. Since AI programs *construct* solutions to problems by simulating concrete task situations, the computer-modeller faces the need for much more detailed and precise analyses than non-computational modellers do. The imperative of getting something to work becomes as important as the simulation, instantiation or discovery of theoretical principles. Given

this, it is easy for computer-modellers to become mired in detail, so that fundamental principles become inseparable from supporting detail [43]. Moreover, the hard work that is done in furnishing the supporting details may tempt one to emphasize performance rather than principles in reports of one's work.

Programs in NLG do not all fit under the category of *AI programs* that have been the subject of methodological scrutiny and debate over the last ten years or so. In NLG, there has been a greater variety of purposes of programming, and a more diverse number of ways of inputting ideas into programs, a most notable method being the adaptation of theories and results from related disciplines like linguistics and psycholinguistics. Perhaps this is reflective of a process by which a gradual refinement of the identity of AI as a field is occurring. Although it is possible to devise metrics for various aspects of an NLG system's performance or potential, it is impossible to carry out an informed assessment of an NLG system without considering its objectives and assumptions. In the reminder of this section, we first discuss two major roles of programming as an activity integral to theory-forming in AI in general and NLG in particular. We then consider programs and theories as evolving entities. Finally, we briefly discuss some issues relevant to scientific interests in NLG.

2.1. The role of programming

We can identify two major roles for programming as a theoretically-motivated activity in AI and cognitive science. They are, namely, (1) programming as a computational analogue of psychological (or scientific) experiments and (2) programming as systematic exploration. Candidates of both kinds exist among NLG systems. A third major kind of NLG systems, primarily motivated by computational-linguistic interests, is discussed in section 3. We may note that in NLG, as in most other branches of AI, motivations can be multiple. Particular projects may not have all of those motivations, and if they do, they may have them prioritized unequally. Moreover, the motivations may not be explicitly spelled out. However, factoring out the possible purposes of implementations and dimensions of potential contribution would greatly facilitate assessing and recording the positive contributions along each axis. Bundy [6] classifies AI research according to motivations for doing AI, and groups methodologies and evaluation criteria according to the classification so obtained.

2.1.1. Programs as experiments

Lehnert [20] draws parallels between experimental methodologies in psychology and AI. Garnham [9] also sees programming in AI as a viable alternative to experimental psycholinguistics. Lehnert points out correspondences in ordered steps as follows:

1. propose computational theory <-----> propose psychological theory
2. implement the program <-----> set up the experiment
3. execute the program <-----> run the experiment
4. analyze program output <-----> interpret experimental results

This analogy, while largely valid, conceals crucial differences between the activities of

implementing the system and setting up the experiment. Program output and experimental results both require a theoretical framework for their interpretation. However, while the *setting up* of an experiment can be carried out in relative isolation from the theory, the activity of system implementation, during its course, is never too far removed from the theory, and can often suggest revisions to the theory [31].

Programs failing to produce certain outputs are compared with experiments failing to produce certain results. Apart from *bugs* which can be *fixed*, there are several ways in which the failure of a program run can be of significance. However, contrary to what Lehnert [20] implies, they need not necessarily warrant a revision of the proposed theory. Lehnert doesn't consider the possibility that the failure could be primarily mathematically significant (due to an inefficient algorithm or an attempt to solve an intractable problem). Moreover, mathematically relevant properties of programs may not always have immediate cognitive interpretation, or worse yet, may not be cognitively relevant at all [3]. Nevertheless, it is generally accepted that a program judged to be successful gives empirical credibility to the theory it embodies. We discuss this type of programs again in section 2.4 with a different perspective, and point out instances of NLG research that come under this category.

2.1.2. Programming as exploration

Programming can be, and has been, quite effective as a means of systematic exploration of new research issues. The investigation could be carried out with a variety of theoretically interesting phenomenon-driven motives, such as examining how a newly-considered independent variable can exercise control over a phenomenon, or, how a newly-considered phenomenon can be modelled. Alternatively, the exploration could be intended to test the efficacy or usability of a new formal tool (representational language, control strategy, architectural design, etc) or a programming tool, or finding a new use (that is, in NLG) for an old tool found effective in other applications.

This type of activity has been a fairly fertile source of results in NLG. Whether the research is primarily motivated by the phenomenon or the tool, the criteria of scientific model-validation are not applicable to these cases. Nor do such systems make scientific claims, though the exploration could be tempered by systematicity and rigour. Programming of this kind also helps secure insights that introspection alone cannot provide. It enables one to concentrate on aspects that non-computational theories miss, and furnishes theories with the necessary supporting detail [41]. We summarize below a few major representatives of this type of effort in NLG. It may be noted that they overlap with the realm of computational linguistics too.

1. Identifying and incorporating a new linguistic or discourse aspect. Such work is frequently correlated with exploration of new domains, settings or genres of text, in which the domain details can be seen to be determinants of the aspect of interest. Exploration of this kind will continue to be valuable, since it is impossible to observe all facets of natural language in one domain or setting. However, exploring new domains is just one way of

discovering new problems. Example: r-expression generation in recipes (Dale [7]): the *EPICURE* system

2. Identifying and taking into account an independent variable as a knowledge source. Justification for isolation of the component as a knowledge source is given by one or more of linguistic, discoursal, psychological, computational or practical (usefulness in applications) considerations. The knowledge source is used in text planning, realization, lexical selection or other NLG subprocesses. Example: user model in generating object descriptions (Paris [34]): the *TAILOR* system

3. Exploration of certain processing strategies in NLG. Example: restrictive and prescriptive planning in multisentential text generation (Hovy [12]): *PAULINE*

4. A NLG subprocess in a specific (perhaps real-world) task/application setting. Example: text planning for advisory dialogues (Moore & Paris [32])

5. Algorithm testing: implementations of newly designed algorithms. Example: reversible grammar to invert unification parser into generator (Peng & Strzalkowski [37])

6. Demonstrating suitability of a formalism/tool for an NLG task. Example: RST (Rhetorical Structure Theory) in planning coherent paragraphs (Hovy [13])

2.2. Theories and programs as evolving entities

In this subsection we reconsider the relation between programs and theories from the point of view of their evolution: the ways in which programming and theory formation overlap temporally and influence each other. Programs as well as the theories they instantiate go through a process of successive refinement or approximation towards their final form. Programs progress from fragility to robustness, from inefficiency to efficiency, from a tentative design to a principled design, or from disparity to fidelity of simulation. Theories progress to higher levels of generality, empirical validity, modularity, economy of expression, coverage, elegance or other virtues. Sufficiency conditions are tightened into more persuasive necessity conditions through a process of *competitive argumentation* [43].

The time-scales for reaching the goals, intermediate or final, can be fairly large. Researchers are confronted with different kinds of obstacles at each stage. The initial venture into a task domain to obtain first formalizations or statements of near-principles can be no less challenging, and is no less important, than the ultimate stages of model validation.

In the research literature on NLG, we come across reports of theories and programs at diverse states of evolution or maturity, and frequently, reports of the same theory or program at different points of its evolution. Different tools are necessary or appropriate to propel the theory or program to the next higher stage. Different sets of criteria are

appropriate to evaluate the theory or program at any given stage. As has been pointed out by many researchers such as Lehnert [20], VanLehn *et al.* [43] and Wilks [44], computational theories of cognition (natural language generation included) are best seen as engineering-style *successive approximations*. Contrary to the norm in natural sciences, neither claims to modelhood of nascent work on the part of researchers, nor binary validity judgments on the part of evaluators is appropriate for relatively young computational-cognitive disciplines of enquiry.

Miller *et al.* [31] mention three general stages through which a model passes, and discuss ways in which (psychological) experiments can be useful ancillaries at each stage. We transfer the discussion analogically to programming, as follows. A vast majority of NLG programs come under (1) and (3).

1. *Initial theory formulation*: During the initial stages of formulation of a theory or study of a problem, programming (for example, implementing a *prototype system*) can suggest if the ideas are viable, on the right track, etc, and can help assure the confidence of the researcher and the research community in the course and value of the project.

2. *Explicit models*: Models, at this stage, can make quantitative predictions on data. Programs can help fit model to empirical data, or to choose from among competing models, given data. At this stage, the level of advancement of research should be sufficient to guarantee consensus about what concepts, findings and assumptions are critical to the problem, and define the goals of evaluation precisely.

3. *Middle ground*: Here, the components of the theory have been spelled out in sufficient detail to support instantiating it into a program. Programming and theory formation proceed interleaved, and are likely to influence each other. Evaluation can help decide whether the implementation is a good instantiation of the theory, and whether a certain feature of the program is theoretically essential, implementationally essential or a (justifiable) kludge. Such evaluation is very often carried out by the researchers themselves as a self-check measure when they develop their work.

2.3. Scientific Issues

The diverse views in AI and cognitive science on scientific quests and claims have been of the sort:

- can do science, this way
- must do science
- can't do science
- can't do science; not now at least
- can do science, but not in this paradigm

This diversity stems partly from the variety of original academic traditions of researchers doing AI or cognitive science and the correspondingly-different notions of what constitutes a scientific theory. From the discussions we extract some key concepts relevant to moulding scientific theories in NLG and present them briefly in this subsection.

Models of scientific theories should be *predictive*, that is, able to display or explain phenomena not explicitly incorporated into the theory *a priori*. In discussing computer simulation models, Bratley *et al.* [4] state that a model that merely accommodates known facts by incorporating them into itself is an ad-hoc fit, and that a worthwhile model must generate predictions, and thereby, new insight. Not all *theories* need to *predict*: Wilks [44] observes that AI theories can be theories of several things at once, just as several descriptions can be theories of the same subject matter at the same time. All that is required is that the statements of a theory should be logically related to each other and to the subject matter. However, Wilks points out that a *scientific theory*, by definition, should promise predictions.

Also essential to scientific theory forming is the notion of *falsifiability* or *refutability*. There is no such thing as a *correct* scientific theory; only *not-yet-falsified* scientific theories exist. Nor is it a discredit for a theory of today to be falsified later. Amenability to examination and falsifiability are essential traits of scientific theories. Similarly, a clear notion of falsifiability and a well-developed understanding of what constitutes crucial falsifying data are hallmarks of scientific methodology.

Another property of scientific theories pertains to the relation between the principles of the theory and the data (subject matter). VanLehn *et al.* [43] distinguish between "*deep*" theories and "*shallow*" theories (*deep* and *shallow* are not used in their normative, literal sense; hence the quotes). "*Deep*" scientific theories employ many layers of unobservables that are remote from data. "*Shallow*" theories attempt to stick as close to observables as possible. In "*deep*" theories, the relation between principles and data is often indirect. The derivation of empirical predictions from the principles can be quite complex. It can be very difficult to assign empirical responsibility to a single component of the theory. Computational theories of NLG (and other departments of human cognition) are "*deep*" because they employ mechanisms and representations that are unobservable and remote from data.

The role of programs as experiments (discussed in section 2.2.1) largely overlaps with the view of programs as *models* of scientifically-oriented theories. Programming is the activity of constructing a model. The model is expected to possess certain static (design) and dynamic (run-time) properties which are functionally isomorphic to corresponding properties in the problem being studied. Getting the program to run successfully is considered tantamount to validating the simulation model. In this regard, we may first recall that researchers have expressed objections to proposing fresh work as scientific theories [44], although such work can quite possibly *lead to* the gradual formation of scientific theories. We may also recall the tendency to view computational-cognitive theory formation as a process of successive approximation rather than as a one-time process subject to immediate success/failure judgment. Moreover, programs are generally not in themselves accepted as theories or as empirical proofs of theories ([39], discussed

in section 2.1) unless a well-articulated statement of the theory separate from the program is furnished. With these points in mind, we recognize a class of NLG systems/theories as *scientifically-oriented* ones (rather than as *scientific* ones):

They aspire to be process models, or scientific computational theories of NLG. They may generate a restricted type of text, or may deal with restricted domains. Such research can be of relevance to related disciplines like psycholinguistics, especially in terms of what they reveal about the *architecture* of the generator and about *control* mechanisms. Systems of this kind may adopt various theories and formalisms from related disciplines. However, if the enterprise is one of scientific theory forming, then the choice of tools, formalisms and frameworks is ideally subordinated to the problem, rather than the other way around. Here are some eminent examples in NLG:

- Description directed control, executable descriptions in deterministic generation (McDonald [25], [24]): the early versions of the *MUMBLE* system
- Bounded parallelism, common program hypothesis, dependency propagation (Hasida [11], Hasida & Isizaki [10])
- Incremental procedural grammar (*IPG*) (Kempen & Hoenkamp [19])
- Uniform architecture for parsing and generation (Shieber [40])

In concluding this section, we consider briefly the notion of *levels of adequacy* of scientific theories. The levels of adequacy correspond to a ranking of the depth of research questions asked. Roughly put, *observational adequacy* corresponds to relatively straightforward questions, *descriptive adequacy* to intermediate-level questions and *explanatory adequacy*, to deeper questions. The diversity of research efforts in NLG is also manifested in the diversity of depths of questions asked and levels of adequacy aspired for by the theories. A plausible explanation for the perceived confusion or the lack of structure in the field as a whole is that questions pertaining to all levels of adequacy are being pursued simultaneously. It is perhaps not advisable to legislate an arbitrary research program with a certain temporal agenda as *the* agenda for the entire NLG community.

3. Some Computational-Linguistic Issues in NLG

In this section we focus on the representation of linguistic knowledge in NLG systems/theories. Viewing language from the perspective of how utterances are produced has led to the development of novel notions of grammar and given prominence to functional theories of language. The development of metrics and procedures for the comparative evaluation of the syntactic aspects of NLG systems is complicated by the widely differing standpoints taken on grammaticality, structure derivation and syntactic knowledge.

In the previous section, we discussed two major types of theoretically-motivated NLG systems, viz., the exploratory systems and the scientifically-oriented systems. In this

section we introduce a third major type of NLG systems: these systems are computational-linguistically motivated, and are built primarily to implement, adapt or demonstrate the implementability of various formalisms or descriptive theories from related disciplines like linguistics and philosophy of language, or to investigate linguistic knowledge in NLG systems with a psycholinguistic perspective. Under this category we include not only single-sentence systems working in minimal contexts, but also systems generating discourse (multisentential monologue, dialogue, etc). Some of the theories may have been stated axiomatically (for example, GB-theory), and the system construction in this case explores how a procedural slant can be given to the theory, and how the theory could be used in NLG. Depending on how well-worked-out the computational aspects of the original theory are, the implementation effort may be mainly an exercise in furnishing the principles of the theory with technical detail (this is *not* a trivial task), or re-interpreting the principles of the theory to give them procedural import. However, the technical difficulties encountered during the implementation can reveal shortcomings in the theory being implemented, and can suggest revisions to the original theory.

It is very difficult to obtain a cohesive classification that encompasses all NLG systems of this type. We propose to orient this discussion along the lines of the following three questions on grammatical knowledge:

- where does it reside?
- what factors are deemed essential to grammar?
- how is it embodied in NLG systems?

3.1. Where does grammatical knowledge reside?

This question is tied up with the role of the lexicon in NLG. Most generators modelled after traditional linguistic theories had a passive lexicon which was a mere database of words. They entrusted structure-building to a syntactic component, typically represented as a collection of phrase-structure rules. However, contemporary perspectives in NLG as well as in computational linguistics and formal syntax view the lexicon as an active component in syntactic structure-building. In NLG, this position is evident in *IPG* [19], as well as in systems using formalisms/theories like HPSG, LFG and Meaning-Text Theory. Certain other systems view the lexicon as the repository of other kinds of knowledge too. For example, in NLG systems implementing Meaning-Text theory, such as those of Iordanskaja *et al.* [17], lexical knowledge also includes conceptual (world) knowledge.

3.2. What factors are deemed essential to grammar?

This question is related to the fuzzy boundaries between syntax, semantics and pragmatics, and the many related unresolved issues. While a typical view of what constitutes syntactic knowledge might include only *categorial/phrasal* notions, others also include *grammatical-functional* notions like subject, object, etc, that provide connection to semantics.

In certain other systems, *communicative-functional* notions, that provide connection to pragmatics, are also considered very much integral to linguistic knowledge. A very prominent example of such a linguistic theory is systemic grammar, and its representative in NLG, the *Nigel* grammar formalism [22], and the *Penman* system [23]. Another system that keeps linguistic knowledge tightly coupled to language function is the *TELEGRAM* formalism in Appelt [2] which focusses on the *intentional-function* of language. Using the theoretical paradigm of *action planning*, Appelt views both domain-level tasks and utterance production (linguistic) tasks uniformly as *goal-directed actions*, and NLG, as *speech-act planning*.

Since conditions of language use are temporally prior to the choice of linguistic constructs in NLG, it seems natural that NLG researchers, who have to specify the pre-linguistic conditions that lead to the linguistic choices, have been interested in such theories as systemic grammars and speech-acts theory.

3.3. How is grammatical knowledge embodied in NLG systems?

NLG systems differ widely on how they represent grammatical knowledge. Most discussions on this issue in computational linguistics revolve around the comparison of principle-based approaches with rule-based approaches. However, from the point of view of NLG, the question is much broader. In order to get a good grasp of the differences among NLG systems on the question of representation of grammatical knowledge, we pivot the discussion on the *competence* vs *performance* distinction.

The various shades of distinction that have been correlated with the *competence* vs *performance* distinction include *potential* vs *performance*, *knowledge representation* vs *knowledge use*, *what the system* **can** *do* vs *what the system* **does** *do*, *database* vs *processor*, etc. This discussion is not intended as a protracted exposition of *competence* vs *performance*. As samples of good discussions on the legitimacy, relevance, usefulness and implications of the distinction, we cite [44], [43], [42] and [8] and some of the references cited therein.

We note that both competence-oriented as well as performance-oriented theories of language have been in vogue in NLG. Since NLG systems are performance models, we focus on the disparity between performance and competence as manifested in NLG systems, and propose to view the systems under the following three-way classification:

- Type-*C*: Competence is prior. Performance is construed as the use of competence. Linguistic knowledge is explicitly represented as a competence-characterization (neutral between generation and analysis), and performance is implemented by retrieving the knowledge through appropriate general access mechanisms which are devoid of any linguistic knowledge in themselves.

- Type-*P*: Performance is prior. Competence is implicit in performance. The theory/system may include assumptions about competence; however assumptions and hypotheses about performance are crucial. They primarily

seek to explain or model performance, rather than to find ways of justifying the relevance of competence theories to performance, or to implement competence theories as theories of performance.

- Type-*CP*: These theories seek to explain the complex connections between performance theories and competence theories in systematic ways. Assumptions, data and results regarding performance as well as competence are important.

3.3.1. Type-C Systems

Type-*C* NLG systems include *rule-based* approaches, in which the knowledge about structure-derivation is stored in rules which are operationalized by means of a rule-interpreter. Alternatively, in *principle-based* approaches, the knowledge is represented in declaratively-specified constraints. The latter approach to describing language characterizes axiomatic theories like GB-theory. Arguments favouring principle-based approaches to NLG are commonly stated in terms of such desirable properties as modularity, declarativity, concision and high degree of parameterization of the principles in the theories they embrace (for example, [1]). However, much as the theory may provide the NLG systems with linguistic well-foundedness, in practice, it may not readily lend itself to implementation in NLG systems first, the NLG system implementer will have to tackle the details of such missing components of the theory as semantics and lexicon and the necessary interactions with these components. This, however, is a common challenge faced by all attempts to employ theories in contexts broader than the theories' central concerns.

A different kind of challenge consists in interpreting the principles of the theory suitably for computer implementation. For example, principles stated as filters, applied to sentences after their generation in the theory will have to be re-interpreted as active constraints that apply at a prior point and thereby prevent generation of ill-formed sentences in the NLG system. The term *generation* in *generative grammar*, *deep generation*, etc is meant in an abstract sense, and is not to be mistaken with the subject matter of our concern, viz., *natural language generation*. It is misleading to interpret the terms of the theory as providing a direct blueprint for implementing its principles in models of NLG.

As examples of type-*C* implementations we may cite experiments with GB-theory in [5] and [1]. Also included under this category are implementations using non-transformational formalisms like GPSG, LFG and HPSG, which have been formulated in terms more easily amenable for computer-processing, using *unification* as an essential processing mechanism.

3.3.2. Type-P Systems

These systems overlap with the scientifically-oriented theories of NLG mentioned in section 2.3. In these systems, certain psycholinguistic observations about human language production, including empirical and experimental data on speech errors, are

used to theoretical and computational advantage. Linguistic knowledge is encoded in these systems in pre-compiled form. Inference computation at run-time is reduced to a look-up or direct retrieval, thereby giving impressive gains in efficiency. Such an approach is also applied in other aspects of cognition like knowledge-representation and reasoning (for example, [18]). The degree of pre-compilation may vary between systems. IPG [19] and early versions of *MUMBLE* [25, 24] fit this category. They both consider the possibility of linguistic knowledge being resident in the architecture and the control mechanisms of the generator in theoretically-significant ways.

3.3.3. Type-CP Systems

These are similar to type-*P* systems in their scientific orientation and the use of psycholinguistic data and compiled knowledge. However, while type-*P* systems do not deal with how the compiled form of linguistic knowledge is obtained (perhaps from more fundamental specifications), type-*CP* systems attempt to unravel the connection between performance (NLG) models and fundamental specifications in principle-based approaches like GB and HPSG. We cite the work of Hasida [11, 10] as the only instance that we know of this category. The works of Hasida and Kempen and Hoenkamp have been articulated in single-sentence contexts. *MUMBLE* has been applied quite successfully in multisentential contexts. The main difference between type-*C* and type-*CP* systems can be stated as follows: type-*C* systems aim at a more-or-less direct realization of the principles of a linguistic theory in NLG. Type-*CP* systems, by grounding their theory in psycholinguistic hypotheses and data on performance in NLG, aim at a more subtle interpretation of the principles.

3.3.4. Other Issues

There are linguistic theories, like systemic grammar, to which the *competence* vs *performance* distinction is not critical. These theories typically do not just consider language as a psychological faculty, but also take into account the properties of language use in context. NLG systems instantiating such theories (for example, [36] and [23]) share characteristics in common with both type-*C* and type-*P* systems.

There is a host of other questions that we have not touched upon which are just as important to our concerns: in particular, issues of text planning, multisentential paragraph and dialogue generation, and generator efficiency. As samples of thought-provoking discussions and comprehensive surveys, we point to Hovy [15] for paragraph planning, McDonald *et al.* [28] for architectural and efficiency issues, and McKeown & Swartout [29] for explanation generation.

4. Evaluating NLG systems

Evaluating NLG systems may involve factors that are quite nebulous (and hard to quantify), but probably not altogether intangible. We may also note two distinct facets of evaluating NLG systems, for each of which evaluation dimensions and criteria can be formulated separately:

- evaluating in isolation, for the system's own merits and deficiencies

- comparative evaluation

In this section we give a brief summary of some factors affecting evaluation of NLG systems and a list of criteria relevant for evaluating NLG systems/theories.

4.1. Factors Influencing Complexities of Evaluation

Here we summarize our observations on some of the factors that pose challenges to the task of comparing and evaluating NLG systems, dually viewed as theories.

- Diversity of objectives, assumptions and methods of system building (programming), phenomena of interest and research-guiding questions, as described in sections 2 and 3. Understanding them is crucial to making the evaluation not only *accurate* but also *fair*.

- Variety in delineation and modularization of subtasks.

- Diversity in the form, content and structure of mental models, protosemantic knowledge and the input to the generation process. Though all NLG systems *end up* alike, with natural language output, their starting points are different, dependent on the conceptual domain and the genre of text, and are in most cases hard to verify or justify in terms of empirical or experimental findings. This factor affects the text planning stage of the NLG system most adversely, since different text planners will have to do different kinds and amounts of preparatory work to synthesize messages or r-specs for the realizer to generate surface text.

- Variation in the choice of knowledge representation schemes and other modelling tools. The difficulty in evaluating NLG systems may be related to the inherent difficulty in drawing equivalences between formal languages.

- Difficulty in defining *cognitive adequacy* consensually. Intuitions can clash, and *armchair-psychology* may not be adequate. Although the theories we form are *computational*, the subject matter of concern is a *psychological* process, and hence, psycholinguistic findings should be taken into account to the extent that they are available and relevant. However, for many phenomena that interest NLG researchers, psycholinguistic findings giving sufficiently useful evidence are not available. Even psycholinguistic research has been biased by a preference for language understanding, for the purely practical reasons of the difficulty of setting up controlled experiments on language production [9]. As in most sciences, a symbiotic relation is desired between theory and experiment. The inability of experiments to support theory formation may be due in part to the lack of a theory to interpret the experiments.

- Often, the programming vehicle or the environment chosen can present additional obstacles. For instance, the entire implementation may be carried out in a uniform representational and programming system like LISP or Prolog. However, the abstraction facilities of the chosen system, while being more naturally suited to the NLG abstractions at some stages or subtasks of the model, could be less suited or even ill-suited to implement certain other aspects. It is not uncommon for researchers to feel the need for the best of all the programming tools and environments! The evaluation of NLG systems will have to consider carefully whether the shortcomings and limitations are in the essence of the theory itself, or in the incidental, ad-hoc working

assumptions and appendages that have to do with the chosen implementational vehicle.

4.2. A partial list of evaluation criteria

In this subsection we give a list of criteria relevant for the evaluation of NLG systems/theories, pointing, where appropriate, some precautions against misusing them. This list is far from exhaustive. Many of the items in the list have been commonly accepted and used implicitly. It is important to recognize the possible ways in which the various dimensions of evaluation could be related to one another. The possible relations are:

1. *overlap*: Can score well in both at a single shot. For example, coherence and naturalness of text.
2. *independence*: Possible to score well in each one, but work must be done separately to achieve the scores. For example, syntactic coverage and applicability across different domains.
3. *conflict*: Tradeoffs. Cannot achieve high scores in both. For example, tradeoff between between efficiency and flexibility when using precompiled knowledge.

We may also note that discussions of evaluation have not only involved assessment of programs and theories, but also the estimation of overall value or contribution of a program or a theoretical effort. The latter is often not immediately evident upon proposal and publication of the work, may take time to become tangible and measurable, as the field of enquiry progresses, and may defy objective definition.

Here is a small list of criteria relevant for evaluating NLG systems or their modules. Evaluation procedures for *AI systems* have been discussed in [39].

- expressibility: is everything planned realizable? [30]
- efficiency: in general related to generating text with the smallest, or limited amounts of resources (space and time).
- modularity: the cutting-up of the NLG process into modules or stages must be well-motivated [26].
- quality of text-output: coherence. Devices of coherence contribute to efficiency, and hence, to the naturalness of text.
- complexity of task domain/setting: This can be a deceptive yardstick in judging NLG systems. The choice of simple domains or settings is sometimes necessary. Often, the first attempts at eliciting theoretically-important principles are very difficult, and the complexity of task domain details can actually interfere with complexity of argumentation for theoretical principles [43]. It may not always be useful to compare two arbitrary NLG systems with respect to domain complexity. *Equating* complexity of tasks and domain details with theoretical merit is common in cognitive science and AI [43], but in NLG, probably not wise in all contexts.
- complexity of text produced. This too can be a misleading one. Fairly complex text can be produced with unprincipled methods, while systems

producing simpler text can be valuable in terms of their theoretical import.

- extensibility. A system can be extended along several dimensions, some of which are enumerated below. While extension along some of the axes may involve relatively simple incrementation or modification of knowledge sources, extensions along other axes may necessitate architectural extensions, system re-design or quite possibly, revision of the underlying theory.
    - mode: speech vs writing
    - type: monologue vs dialogue or polylogue
    - single sentence to multisentential text
    - to different conceptual domains
    - to different genres of text
    - to other languages

  The same genre of text can be found in different conceptual domains. For example, cookery-recipes, route-prescriptions and instructions for drawing geometry constructions can all be seen as action-prescription text, while each one deals with a different conceptual domain.
- parsimony and simplicity: lack of unjustified redundancies
- predictive value: ability to display or explain new phenomena.
- tailorability: ability to tune the model to account for individual variations [43]. The tuning range of such a model should be well-limited; a model which can be tuned to match almost any data is vacuous.
- empirical coverage: for NLG systems, this includes syntactic coverage, which has been the focus of recent interests in evaluation. Hovy [16] discusses the evaluation of generators for syntactic coverage.

4.3. Other Discussions on Evaluation

Evaluation has also been discussed recently in the context of natural language processing (NLP) systems. Palmer *et al.* [33], reporting on the proceedings of a workshop, centre their general discussion around defining *blackbox* and *glassbox* evaluation, and specifying criteria for the same for NLP systems. They recognize glassbox evaluation as a necessary step preceding blackbox evaluation. Blackbox evaluation is seen as the ultimate goal, from the point of view of making commercial products out of NLP systems. Reciprocal concerns with regard to NLG systems were expressed at a workshop in AAAI'90, and Hovy [16] proposed and discussed procedures for measuring and comparing generators' inputs and outputs.

Much of the interest and effort has been focussed on one aspect of the systems, viz., the quantitative estimation and comparison of the performance of the systems in terms of their grammatical coverage (*test sentences* serving as inputs for NLU systems or possible outputs of NLG systems). Just as this paper aims to be pertinent to those interests, on-going efforts on devising procedures for performance evaluation or syntactic coverage

estimation are also integral and useful to the concerns of evaluation in the broader sense of this paper. Theory evaluation does involve performance evaluation [38]. Performance (input-output behaviour) is critical for claims to modelhood of all simulation systems [4].

However, *evaluation* construed in a narrow sense can be misleading if its place in the overall context of the field of enquiry is not fully understood or acknowledged. Blackbox evaluation is most effective and useful when it is conducted in the broader context involving due consideration of theoretical concerns, some of which we have discussed. To be able to measure or automate measurement of the facets of a program's behaviour is certainly a vital aspect of evaluation; the other equally crucial factor is evolving an understanding of how such measurements can be used in arriving at informed judgments. The latter aspect requires a fuller cognizance of how performance evaluation of the systems is often tantamount to theory- or methodology evaluation.

Moreover, total focus on performance evaluation to the exclusion of theory examination could engender a sense of competition rather than mutual co-operation. This, we feel, is detrimental to research if the evaluation procedure directs the efforts of the researchers to meet the goals of evaluation by imposing artificial, theoretically-unsound constraints: performance so achieved will be a spurious measure of progress in the field. Besides, in a relatively-newly-emerged field of enquiry like NLG, evaluation methodologies should guard against wittingly or unwittingly sponsoring particular competitors (and excluding others out of reckoning) by giving them unfair advantage due to hidden presumptions or biases, when consensus on critical theoretical issues has still not been reached, and approaches are plural. In fact, the AAAI'90 workshop on NLG system evaluation has been very valuable in raising an explicit discussion and awareness of these issues.

5. Conclusions

We have briefly discussed some of the major influences that have shaped NLG as a field of enquiry in terms of the variety of research objectives and methodologies. These influences have correspondingly manifested themselves in the approaches to theoretical issues and in the roles of programming NLG systems or subsystems. If we may invoke the legendary *what-how* distinction that is native to NLG, we have not furnished detailed evaluation procedures (the *how* of evaluation), but have focussed instead on the possibilities of 'inputs' that evaluation procedures should take into reckoning (the *what* of evaluation) in terms of some significant purposive, methodological, metatheoretical and theoretical dimensions. In so doing, we have frequently viewed the prospective object of evaluation dually as a system and as a theory.

Pluralism of approaches could be perceived as *methodological mess* [35], but it is there, and is necessary [31]. At the expense of adding to methodological confusion, open-mindedness has enabled transfer of valuable insights. All of the mainstream methodologies of the disciplines influencing NLG have the capability of making

important contributions to computational research on NLG. The trick is in pluralizing without losing identity. Identity of NLG as a field, we believe, comes from computational orientation.

The scopes of the research efforts are typically circumscribed, and simplifying assumptions made accordingly, so as to facilitate exploration of the issue at hand. Insights about syntactic structure derivation are gained by neglecting semantic and lexical factors; lessons on single-sentence generation are learned in the absence of discourse considerations, and so on. Nuances like metaphor, metonomy, compound nouns and conversational implicatures, while negligible or intrusive for some researchers, are veritable objects of quest for some others. In attempting to integrate results from one theory or project into another, one faces difficulties even when the major assumptions and approaches are shared. Results may not be integrable wholesale: a lot of adaptation work is necessary. Sometimes, by broadening the scope of the enquiry, the lessons learnt from narrowly-focussed elegant theories are challenged, and the legitimacy of the neglect of a variable, questioned. Attempting to explain all aspects of NLG in one theory is an overwhelming task. Not all premises are relevant to all research questions. We will have several partial theories with common threads of assumptions or results running through them. Extending the capabilities of an NLG system, in addition to programming extensions, often involves theoretical extensions.

Although NLG is not quite an *emerging* field any longer, it is still younger than NLU and AI. The identity of the field has been sufficiently established, and a solid base built, with a number of impressive results in both theory and programming. Unexplored territories of the problem still abound. However, this is high time efforts are also directed at identifying similarities and differences between hitherto-disjoint and uncompared results. Such efforts to unify, in addition to improving cohesion in the discipline and consolidating its identity, will reveal additional research problems and fresh insights. The current interest in evaluation is precisely a step in that direction, as it has catalyzed explicit discussions of what makes the different NLG systems and theories similar or different, and at what levels comparisons and contrasts can be established meaningfully. This is also a stage where research problems are aplenty and maintaining interest in the growth of the field is critical.

The present course of the field appears natural, and the efforts will in time bear fruit in terms of improved theoretical understanding as well as more sophisticated NLG systems, be they research systems or commercial products. When research reaches such a stage where there is sufficient consensus on what the critical results and concepts are, it will be easier to define clearly the goals of evaluation. By sustaining and nurturing interest in the field at large, we should be able to devise evaluation methodologies fairly easily as a natural consequence of refined research methodology and improved theoretical understanding, and do so in ways in which it is beneficial to the research community and

to the genuine furthering of knowledge about NLG.

Acknowledgments

Our thanks are due to the organizers and attendees of the *AAAI'90 Workshop on Evaluating NLG Systems* for thought-provoking questions and comments, to Bob Hadley and Ray Jennings for discussions, and to Ranabir Gupta for his helpful comments on an earlier version of this paper. The first author gratefully acknowledges partial financial support from a Simon Fraser University Graduate Research Fellowship, a British Columbia Advanced Systems Institute Graduate Student Scholarship and Natural Sciences and Engineering Research Council of Canada operating grant A-4309.

References

1. Amsili, P. and Saint-Dizier, P. *A Generation Method Based on Principles of Government and Binding Theory*. LSI Universite Paul Sabatier, 1990. Manuscript obtained from the authors.

2. Appelt, D.E. *Planning English Sentences*. Cambridge University Press, 1985.

3. Berwick, R.C. and Weinberg, A.S. "Parsing Efficiency, Computational Complexity and the Evaluation of Grammatical Theories". *Linguistic Inquiry 13*, 2 (1982), 165-191.

4. Bratley, P., Fox, B.L. and Schrage, L.E. *A Guide to Simulation*. Springer-Verlag, 1983.

5. Brown, C., Pattabhiraman, T. and Massicotte, P. Towards a Theory of Natural Language Generation: The Connection between Syntax and Semantics. In *Natural Language Understanding and Logic Programming, II, Dahl, V. and Saint-Dizier, P., Editors*, North-Holland, 1988, pp. 239-254.

6. Bundy, A. What Kind of Field is AI? In *The Foundations of Artificial Intelligence: A Sourcebook, Partridge, D. and Wilks, Y., Editors*, Cambridge University Press, 1990, pp. 215-222.

7. Dale, R. *Generating Referring Expressions in a Domain of Objects and Processes*. Ph.D. Th., University of Edinburgh, 1988.

8. Fromkin, V.A. "Speculations on Performance Models". *Journal of Linguistics 4* (1968), 47-68.

9. Garnham, A. *Psycholinguistics- Central Topics*. Methuen, 1985.

10. Hasida, K. *Bounded Parallelism: A Theory of Linguistic Performance*. Ph.D. Th., University of Tokyo, Department of Information Science, 1985.

11. Hasida, K. and Isizaki, S. Dependency Propagation: A Unified Theory of Sentence Comprehension and Generation. Proceedings of the 10th IJCAI, Milan, 1987, pp. 664-670.

12. Hovy, E.H. Two Types of Planning in Language Generation. Proceedings,26th Conference of the ACL, Buffalo, 1988.

13. Hovy, E.H. Approaches to the Planning of Coherent Text. Presented at the Fourth International Workshop on Natural Language Generation, Los Angeles, 1988. Revised Manuscript, 1989.

14. Hovy, E.H. *Private communication: description of relevant questions for the AAAI'90 Workshop on Evaluating Natural Language Generation Systems*. USC/Information Sciences Institute, 1990.

15. Hovy, E.H. Unresolved Issues in Paragraph Planning. In *Current Research in Natural Language Generation, Dale, R., Mellish, C. and Zock, M., Editors*, Academic Press, 1990, pp. 17-45.

16. Hovy, E.H. Initial Ideas on a Framework for Evaluating Generators. Presented at the AAAI-90 Workshop on the Evaluation of Natural Language Generation Systems, 1990, pp. Boston.

17. Iordanskaja, L., Kittredge, R. and Polguere, A. *Implementing a Meaning-Text Model for Language Generation*. Presented at the 12th COLING, Budapest, 1988.

18. Johnson-Laird, P.N. *Mental Models: Towards a Cognitive Science of Language, Inference and Consciousness*. Cambridge University Press, 1983.

19. Kempen, G. and Hoenkamp, E. "An Incremental Procedural Grammar for Sentence Formulation". *Cognitive Science 11* (1987), 201-258.

20. Lehnert, W.G. Paradigmatic Issues in Cognitive Science. In *Method and Tactics in Cognitive Science, Kintsch, W., Miller, J.R. and Polson, P.G., Editors*, Lawrence Erlbaum, 1984, pp. 21-49.

21. Lenat, D.B. and Brown, J.S. "Why *AM* and *EURISKO* Appear to Work". *Artificial Intelligence 23* (1984), 269-294.

22. Mann, W.C. Nigel: A Systemic Grammar for Text Generation. Tech. Rept. ISI/RR-83-105, USC/Information Sciences Institute,1983.

23. Mann, W.C. An Overview of the PENMAN Text Generation System. Proceedings, AAAI'83, 1983, pp. 261-265. Also appears as USC/Information Sciences Institute RR-83-114.

24. McDonald, D.D. *Natural Language Generation as a Process of Decision-Making under Constraints*. Ph.D. Th., Massachusetts Institute of Technology, 1980.

25. McDonald, D.D. "Description Directed Control: Its Implications for Natural Language Generation". *Computers and Mathematics 9*, 1 (1983), 111-129.

26. McDonald, D.D. Modularity in Natural Language Generation: Methodological Issues. Proceedings of the AAAI Workshop on Text Planning and Realization, 1988, pp. 91-98.

27. McDonald, D.D. Establishing a Common Terminology: Possibilities and Pitfalls. Presented at the AAAI-90 Workshop on the Evaluation of Natural Language Generation Systems, Boston, 1990.

28. McDonald, D.D., Vaughan, M.M. and Pustejovsky, J.D. Factors Contributing to Efficiency in Natural Language Generation. In *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics, Kempen, G., Editor*, Martinus Nijhoff, 1987, pp. 159-181.

29. McKeown, K.R. and Swartout, W.R. Language Generation and Explanation. In *Advances in Natural Language Generation- An Interdisciplinary Perspective. Vol. 1, Zock, M. and Sabah, G., Editors*, Ablex Publishing Corporation, 1988, pp. 1-51.

30. Meteer, M.W. Defining a Vocabulary for Text Planning. Proceedings of the AAAI Workshop on Text Planning and Realization, 1988, pp. 115-122.

31. Miller, J.R., Polson, P.G. and Kintsch, W. Problems of Methodology in Cognitive Science. In *Method and Tactics in Cognitive Science, Kintsch, W., Miller, J.R. and Polson, P.G., Editors*, Lawrence Erlbaum, 1984, pp. 1-18.

32. Moore, J.D. and Paris, C.L. Planning Text for Advisory Dialogues. Proceedings of the 27th Annual Meeting of the ACL, Vancouver, 1989, pp. 203-211.

33. Palmer, M., Finin, T. and Walter, S.M. Workshop on the Evaluation of Natural Language Processing Systems. Tech. Rept. RADC-TR-89-302, Rome Air Development Center,1989.

34. Paris, C.L. Responding to a User's Domain Knowledge: Incorporating the User's Level of Expertise into a Generation System. Proceedings of the International Computer Science Conference, Hong Kong, 1988, pp. 301-309.

35. Partridge, D. What's in an AI Program? In *The Foundations of Artificial Intelligence: A Sourcebook, Partridge, D. and Wilks, Y., Editors*, Cambridge University Press, 1990, pp. 112-118.

36. Patten, T. *Systemic Text Generation as Problem Solving*. Cambridge University Press, 1988.

37. Peng, P. and Strzalkowski, T. An Implementation of a Reversible Grammar. Proceedings of CSCSI'90, Ottawa, 1990, pp. 121-127.

38. Polson, P.G., Miller, J.R. and Kintsch, W. Methods and Tactics Reconsidered. In *Method and Tactics in Cognitive Science, Kintsch, W., Miller, J.R. and Polson, P.G., Editors*, Lawrence Erlbaum, 1984, pp. 277-296.

39. Ritchie, G.D. and Hanna, F.K. "*AM*: A Case Study in AI Methodology". *Artificial Intelligence 23* (1984), 249-268.

40. Shieber, S.M. A Uniform Architecture for Parsing and Generation. Proceedings of the 12th COLING, Budapest, Hungary, 1988, pp. 614-619.

41. Steedman, M.J. and Johnson-Laird, P.N. The Production of Sentences, Utterances and Speech Acts: Have Computers Anything to Say? In *Language Production, vol. 1: Speech and Talk, Butterworth, B., Editor*, Academic Press, 1983, pp. 111-141.

42. Valian, V. The Wherefores and Therefores of the Competence-Performance Distinction. In *Sentence Processing: Psycholinguistic Studies Presented to Merrill Garrett, Cooper, W.E. and Walker, E.C.T., Editors*, Lawrence Erlbaum, 1979, pp. 1-26.

43. VanLehn, K., Brown, J.S. and Greeno, J. Competitive Argumentation in Computational Theories of Cognition. In *Method and Tactics in Cognitive Science, Kintsch, W., Miller, J.R. and Polson, P.G., Editors*, Lawrence Erlbaum, 1984, pp. 235-262.

44. Wilks, Y. One Small Head: Models and Theories. In *The Foundations of Artificial Intelligence: A Sourcebook, Partridge, D. and Wilks, Y., Editors*, Cambridge University Press, 1990, pp. 121-134.

Centre for Systems Science

• Simon Fraser University established the **Centre for Systems Science** as a provincial centre for advanced study. The objective of CSS is twofold: coordinate and combine advanced systems research at SFU, and encourage application of research results within industry in order to further the technological development of British Columbia.

• CSS has assembled three research groups to streamline its efforts. The orientation of these groups is as follows: intelligent systems, microelectronics, and computer and communications systems. Because of shared research interests, this latter group merges with LCCR (**Laboratory for Computer and Communications Research**) for the publication of technical reports.

• CSS's three groups involve over 80 skilled faculty researchers principally from SFU's Schools of Engineering Science and Computing Science, but also draw substantially on Mathematics, Physics, Geography, Philosophy, Linguistics, Psychology, Communication, and Kinesiology.